A Modular AC Optimal Power Flow Implementation for Distribution Grid Planning

Adrian Hauswirth
Tyler Summers
Joseph Warrington
John Lygeros
Automatic Control Laboratory
Swiss Federal Institute of Technology (ETH)
Zurich, Switzerland

Andreas Kettner
Alain Brenzikofer
Supercomputing Systems AG
Zurich, Switzerland

***Abstract*—We present a computational tool for solving semidefinite relaxations of multi-period AC optimal power flow (OPF) problems. Chordal conversion techniques are used to exploit problem sparsity. Three features set it apart from similar implementations: First, a new, concise real-valued model exploits the problem structure and avoids introducing redundant constraints. Second, a dynamic choice of constraint type improves computation time for grids with extensive radial subgraphs. Third, a modular software design enables the easy integration of additional models for photovoltaic inverters, optimal storage placement, etc.**

Benchmark results indicate that our computational improvements significantly enhance performance compared to a standard implementation. This holds in particular for large-scale networks and power grids with large radial subgraphs. Finally, a case study showcases the potential of our modular OPF software design.

***Index Terms*—Convex Optimization, Optimal Power Flow, Smart Grids**

I. Introduction

Government incentive programs in various countries are leading to a massive expansion of renewable energy sources. However, large amounts of intermittent power infeed at the distribution level are known to jeopardize grid stability. Furthermore, the non-controllable nature of this infeed makes it difficult to balance supply and demand.

Various methods have been proposed and studied to address these challenges. In this paper we advocate the use of nonlinear optimal power flow (OPF) simulations to assess the potential of these mitigation strategies. In order to efficiently solve OPF problems in a multi-period setting with non-linear flow equations we use convex relaxation techniques studied in [8] and recently summarized in [9].

These convex relaxations of AC OPF problems, first proposed in [2], have recently attracted attention because of their potential to provide globally optimal solutions to non-linear OPF problems in polynomial time.

In this paper we present a software framework for solving AC OPF problems in the context of distribution grids. Our implementation introduces two modifications that improve performance and a modular design that can accommodate various power systems components and optimization objectives.

The model underlying our OPF formulation ignores uncertainties, assumes steady-state conditions and full information on load and PV generation profiles. We further consider only balanced loads and work with single-phase equivalent circuits. These assumptions do not always hold in practice, but OPF simulations can provide upper bounds on the effectiveness by giving an optimal solution under ideal conditions.

In Section II we introduce the concept on convex relaxations applied to OPF problems. Then, we summarize the chordal conversion techniques used for exploiting sparsity and introduce performance-enhancing modifications to the standard formulations. Section IV introduces the modular software design and Sections V and VI present benchmark results and a case study on optimal storage placement.

II. Complex & Real Bus Injection Relaxations

Consider a power network consisting of a set of nodes $\mathcal{N}$ of size N. Node 0 denotes the slack node which serves as the phase angle reference. In the following we further assume that this node represents the (unique) substation which connects the feeder to higher grid levels. Every node $n \in \mathcal{N}$ has a voltage phasor V_n for the line-to-ground voltage and power injection $S_j \in \mathbb{C}$. Let V and S denote the respective vectors.

Let $\mathcal{L}$ denote the set of undirected lines. We use the notation $m \sim n$ to denote a line between nodes m and n.

We define the feasible set defined by the bus injection model of the AC power flow equations as

$$\mathbb{W} := \left\{ S, V \;\middle|\; S_j = \sum_{k:j\sim k} y_{jk}^* V_j \left(V_j^* - V_k^*\right) \quad \forall j \in \mathcal{N} \right\}$$

where $(\cdot)^*$ is the complex conjugate and y_{jk} denotes the admittance value for the line $j \sim k$.

The admittance matrix of the network is given as $\mathbf{Y} \in \mathbb{C}^{N\times N}$ with $\{\mathbf{Y}\}_{jk} = -y_{jk}$ if $j \sim k$ and $\{\mathbf{Y}\}_{jj} = \sum_{k:j\sim k} y_{jk}$ and 0 otherwise. It can be used the reformulate

The authors would like to thank Marco Mangani of *ewz Energie* and Marc Eisenreich of *BKW Energie AG*.

the feasible set more compactly as

$$\mathbb{W} = \left\{S, V \,\middle|\, S_j = \operatorname{tr}\left\{\mathbf{Y}_j V V^H\right\} \quad \forall j \in \mathcal{N}\right\}$$

where $\mathbf{Y}_j := \mathbf{Y}^H \mathbf{e}_j \mathbf{e}_j^T$ and $\mathbf{e}_j$ is the canonical unit vector in direction j.

By defining $U_j := |V_j|^2$ and $W := VV^H$ we can equivalently write

$$\begin{aligned}\mathbb{Y}:=\{S, U \,|\, & U = \operatorname{diag} W, \\ & S_j = \operatorname{tr}\left\{\mathbf{Y}_j W\right\} \quad \forall j \in \mathcal{N}, \\ & W \succeq 0, \operatorname{rank} W = 1\}\end{aligned}$$

The common convex relaxation first applied in [2] consists in dropping the non-convex rank constraint in order to get a convex feasible set

$$\begin{aligned}\mathbb{V}:=\{S, U \,|\, & U = \operatorname{diag} W, && (1)\\ & S_j = \operatorname{tr}\left\{\mathbf{Y}_j W\right\} \quad \forall j \in \mathcal{N}, && (2)\\ & W \succeq 0\}. && (3)\end{aligned}$$

These equations can consequently be used in a complex-valued convex optimization problem. This requires a solver hat can handle complex numbers, in particular complex semidefinite cones (such as SeDuMi[1]). Alternatively, a real-valued relaxation can be formulated. For this, we define $\widetilde{W} := X^T X$ where $X = \begin{bmatrix}\operatorname{Re} V^T & \operatorname{Im} V^T\end{bmatrix}^T$. We further introduce

$$\begin{aligned}\widetilde{\mathbf{Y}}_j^R &:= \frac{1}{2}\begin{bmatrix}\operatorname{Re}\mathbf{Y}_j & -\operatorname{Im}\mathbf{Y}_j \\ \operatorname{Im}\mathbf{Y}_j & \operatorname{Re}\mathbf{Y}_j\end{bmatrix} \\ \widetilde{\mathbf{Y}}_j^I &:= \frac{1}{2}\begin{bmatrix}\operatorname{Im}\mathbf{Y}_j & -\operatorname{Re}\mathbf{Y}_j \\ \operatorname{Re}\mathbf{Y}_j & \operatorname{Im}\mathbf{Y}_j\end{bmatrix} \\ \widetilde{\mathbf{M}}_j &:= \begin{bmatrix}\mathbf{M}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_j\end{bmatrix}.\end{aligned}$$

where $\mathbf{M}_j \in \mathbb{R}^{N\times N}$ is defined as $\{\mathbf{M}_j\}_{jj} := 1$ and 0 otherwise.

Hence, a real-valued analog to $\widetilde{\mathbb{V}}$ can be defined as

$$\begin{aligned}\widetilde{\mathbb{V}}:=\{P, Q, U \,|\, & U_j = \operatorname{tr}\left(\widetilde{\mathbf{M}}_j \widetilde{W}\right), \\ & P_j = \operatorname{tr}\left(\widetilde{\mathbf{Y}}_j^R \widetilde{W}\right) \quad \forall j \in \mathcal{N}, \\ & Q_j = \operatorname{tr}\left(\widetilde{\mathbf{Y}}_j^I \widetilde{W}\right) \quad \forall j \in \mathcal{N}, \\ & \widetilde{W} \succeq 0\}.\end{aligned}$$

Note that we have $\dim W = N^2$ while $\dim \widetilde{W} = 2N^2 - N$. This alludes to the fact that the real formulation is in general computationally more expensive to use in a OPF problem than the complex formulation.

Finally, note that in order to be able to reconstruct a voltage vector V from either W or $\widetilde{W}$ the rank of the respective matrix has to be 1. In practice, we evaluate the ratio of the two largest eigenvalues to check whether the matrix is close to being rank-1. We will not further investigate this issue in the following. Instead we simply note that reconstructability has not been a problem in the case study conducted for this paper.

[1] www.sedumi.ie.lehigh.edu

III. Sparse Semidefinite Programming for AC OPF

The computational burden of the convex relaxation grows quadratically with the number of buses. It is however possible to exploit the sparsity of the network structure by applying a theorem about semidefinite matrix completion. This idea has been extensively described in [5] and first applied to OPF problems in [7]. Further investigations into sparse semidefinite programming for OPF have been presented in [10] and [1]. The following graph-theoretic notions help to understand the subsequent exposition:

By *clique* we refer to a fully connected subset of nodes in an undirected graph. A *maximal clique* is a clique that cannot be extended by including any other node. A graph is *chordal* if every cycle of size equal or greater to 4 has a chord (an edge joining two non-consecutive nodes in the cycle). A chordal graph can easily be decomposed into its maximal cliques (using the existence of a perfect elimination ordering; see [5] and references therein). Graphs of power grids are in general not chordal. For this reason it is necessary to add (virtual) edges in order to obtain a chordal extension of the network graph. The common way to obtain a chordal extension of a non-chordal graph is by performing a symbolic Cholesky decomposition on the sparsity pattern. This in general implies the application of heuristic ordering to minimize fill-in, such as *approximate minimum degree* ordering (see [1] and references therein). In the scope of this paper the software package CHOMPACK[2] has been used to generate chordal extensions and perform maximal clique decompositions of network graphs.

Inspired by [1], we further introduce two matrix operators: Let $\Lambda_c(W) := W_{c;c}$ denote the principal submatrix of $W \in \mathbb{C}^{n\times n}$ defined by the index set c (for instance a clique). Hence, $\Lambda_c(W) \in \mathbb{C}^{|c|\times|c|}$. In addition, let $\overline{\Lambda}_c(W)$ return the matrix of size $n \times n$ with all zero components except for the principal submatrix associated with the index set c.

A. Matrix Completion and Sparse Semidefinite Programming

Let G be a chordal graph with associated partial matrix M (i.e. not all entries of M are known). Further let $\mathcal{C}_G$ denote the set maximal cliques of G.

The central theorem first stated in [6] is given as follows: M has a positive semidefinite completion if and only if $\Lambda_c(M) \succeq 0 \; \forall c \in \mathcal{C}$. In other words, if the principal submatrices $\Lambda_c(M)$ associated with the cliques $c \in \mathcal{C}$ are positive semidefinite, then the remaining unknown components can be chosen such that M is positive semidefinite.

Consider the set $\mathbb{V}$. We define an *aggregate sparsity* pattern $\mathbf{M}$ given by $\{M\}_{ij} = 1$ if $\{Y_n\}_{ij} \neq 0$. Due to the definition of the trace-operator the sparsity pattern $\mathbf{M}$ applies also to W and we can think of W as a partial matrix. The application of the above results is immediate. Let $\mathcal{C}_{\mathbf{M}}$ denote the set of maximal cliques of the chordal extension of $\mathbf{M}$. Then we replace $W \succeq 0$ with

$$\Lambda_c(W) \succeq 0 \qquad \forall c \in \mathcal{C}_{\mathbf{M}} \tag{4}$$

[2] http://chompack.readthedocs.org

After replacing Equation (4) for (3) in $\mathbb{V}$, the resulting set cannot be directly incorporated in a standard form semidefinite program (SDP) since various components of W are shared between multiple submatrices $\Lambda_c(W)$. In [5] a conversion method is therefore introduced that is later applied in [1], [7], [10] to recover a standard form SDP.

The fact that most off-the-shelf solvers cannot directly exploit sparsity of positive semidefinite (psd) cones makes it necessary to apply this conversion. The downside is that it introduces additional constraints and thus increases the problem size. For this standard form conversion the principal submatrices are defined as separate psd-matrix variables, that is $W_c := \Lambda_c(W)$. However, linking constraints between components of W_c have to be introduced to account for the shared elements of the submatrices $\Lambda_c(W)$. These can be expressed as

$$\Lambda_{c_1\cap c_2}(\overline{\Lambda}_{c_1}(W_{c_1}) - \overline{\Lambda}_{c_2}(W_{c_2})) = 0 \qquad \forall c_1, c_2 \in \mathcal{C}_{\mathbf{M}}$$

In general it is not necessary to establish linking constraints between every possible tuple of cliques since not all of them share elements with each other. The cliques that share elements are easily identified in the *clique tree* that results from the maximum clique decomposition of a chordal graph.

We define the feasible set $\mathbb{M}$ as result of the conversion technique applied to the set $\mathbb{V}$, that is

$$\begin{aligned}
\mathbb{M}:=\{S, U \,|\, & U_j = \sum_{c\in\mathcal{C}_{\mathbf{M}}} \operatorname{tr}(\mathbf{M}_{j;c} W_c) && \forall j \in \mathcal{N} && \text{(5a)}\\
& S_j = \sum_{c\in\mathcal{C}_{\mathbf{M}}} \operatorname{tr}(\mathbf{Y}_{j;c} W_c) && \forall j \in \mathcal{N}, && \text{(5b)}\\
& \Lambda_{c_1\cap c_2}(\overline{\Lambda}_{c_1}(W_{c_1}) - \overline{\Lambda}_{c_2}(W_{c_2})) = 0 && \forall c_1, c_2 \in \mathcal{C}_{\mathbf{M}}, && \text{(5c)}\\
& W_c \succeq 0 && \forall c \in \mathcal{C}_{\mathbf{M}}\}. && \text{(5d)}
\end{aligned}$$

where the the matrices $\mathbf{Y}_{j;c}$ have to satisfy

$$\mathbf{Y}_k = \sum_{c\in\mathcal{C}_{\mathbf{M}}} \overline{\Lambda}_c(\mathbf{Y}_{j;c}) \qquad \forall j \in \mathcal{N}.$$

The analogous holds for $\mathbf{M}_{j;c}$. We further define $\widetilde{\mathbb{M}}$ as the set obtained by applying the conversion method to $\widetilde{\mathbb{V}}$.

B. Reduction of Linking Constraints

It has been argued in [1] that the use of the real formulation $\widetilde{\mathbb{M}}$ should be discouraged since it introduces more than twice as many equality linking constraints between cliques than the complex formulation. However, most available solvers cannot deal with complex variables.

Therefore, we propose a *pseudo-complex* formulation that can be expressed in purely real variables. It exploits the particular structure of the problem and does not require more equality linking constraints than the complex model.

For this, consider

Lemma 1. *Define the surjective mapping* $\Phi : \mathbb{R}^{2N\times 2N} \to \mathbb{C}^{N\times N}$ *with*

$$\Phi(\widetilde{W}) := \widetilde{W}^{11} + \widetilde{W}^{22} + i(\widetilde{W}^{12} - \widetilde{W}^{21}) \tag{6}$$

where $\widetilde{W}^{11}, \widetilde{W}^{12}, \widetilde{W}^{21}, \widetilde{W}^{22} \in \mathbb{C}^{N\times N}$ *are such that*

$$\widetilde{W} = \begin{bmatrix} \widetilde{W}^{11} & \widetilde{W}^{12} \\ \widetilde{W}^{21} & \widetilde{W}^{22} \end{bmatrix}.$$

Then it holds that

$$\widetilde{W} \succeq 0 \quad \Rightarrow \quad \Phi(\widetilde{W}) \succeq 0.$$

Proof: see Appendix A ∎

By applying Lemma 1 we can avoid forming an explicit real-valued problem. For this consider the general decomposed SDP problem (5). Instead of requiring that $W_c \succeq 0 \quad \forall c \in \mathcal{C}_{\mathbf{M}}$, which is a complex psd-constraint, we introduce new real matrix variables $\widetilde{W}_c \quad \forall c \in \mathcal{C}_{\mathbf{M}}$ and use Lemma 1 to formulate the new pseudo-complex problem given by

$$\begin{aligned}
\mathbb{M}^*:=\{S, U \,|\, & U_j = \sum_{c\in\mathcal{C}_{\mathbf{M}}} \operatorname{tr}(\mathbf{M}_{j;c} W_c) && \forall j \in \mathcal{N} && \text{(7a)}\\
& S_j = \sum_{c\in\mathcal{C}_{\mathbf{M}}} \operatorname{tr}(\mathbf{Y}_{j;c} W_c) && \forall j \in \mathcal{N}, && \text{(7b)}\\
& \Lambda_{c_1\cap c_2}(\overline{\Lambda}_{c_1}(W_{c_1}) - \overline{\Lambda}_{c_2}(W_{c_2})) = 0 && \forall c_1, c_2 \in \mathcal{C}_{\mathbf{M}}, && \text{(7c)}\\
& W_c = \widetilde{W}_c^{11} + \widetilde{W}_c^{22} + i(\widetilde{W}_c^{12} - \widetilde{W}_c^{21}) && \forall c \in \mathcal{C}_{\mathbf{M}} && \text{(7d)}\\
& \widetilde{W}_c \succeq 0 && \forall c \in \mathcal{C}_{\mathbf{M}}\}. && \text{(7e)}
\end{aligned}$$

where $\widetilde{W}_c$ are real-valued psd matrices of twice the size/order of W_c. This makes it possible to use the above problem in a solver that can only handle real psd-cones.

The potential speed up from using the formulation of $\mathbb{M}^*$ instead of $\widetilde{\mathbb{M}}$ results from the fact that Equation (7c) introduces less than half as many equality constraints as in $\widetilde{\mathbb{M}}$ where the corresponding constraints take the form

$$\Lambda_{\widetilde{c}_1\cap\widetilde{c}_2}\left(\overline{\Lambda}_{\widetilde{c}_1}(\widetilde{W}_{\widetilde{c}_1}) - \overline{\Lambda}_{\widetilde{c}_2}(\widetilde{W}_{\widetilde{c}_2})\right) = 0 \quad \forall \widetilde{c}_1, \widetilde{c}_2 \in \widetilde{\mathcal{C}}_{\mathbf{M}}$$

where $\widetilde{\mathcal{C}}_{\mathbf{M}}$ denotes the set of extended cliques that result from applying a chordal extension and decomposition to a real-valued model.

C. Dynamic Choice of Constraint Type

Applying a chordal extension and maximum clique decomposition to a radial grid topology results in cliques of size 2 exclusively. Consequently all of the matrices W_c are of size 2×2. In this case, the psd-constraints can be reformulated as second-order cone constraints, that is

$$\begin{bmatrix} W_{mm} & W_{mn} \\ W_{nm} & W_{nn} \end{bmatrix} \succeq 0 \;\Leftrightarrow\; \left\| \begin{matrix} W_{mm} - W_{nn} \\ 2\operatorname{Re} W_{nm} \\ 2\operatorname{Im} W_{nm} \end{matrix} \right\| \leq W_{mm} + W_{nn}$$

A reformulation of psd-constraints makes sense since soc-constraints are in general computationally advantageous.

This conversion can also be partly applied to meshed grids with significant radial subgraphs. This results in an optimization problem that contains both soc- as well as psd-constraints where the type of constraint is dynamically chosen based on the size of the corresponding clique (i.e. the size of the associated submatrix).

IV. Modular OPF Design

To obtain an extensible implementation we isolate and encapsulate non-essential components of the OPF problem in the sense that we group optimization variables, constraints and individual terms into *OPF elements*.

In this context we consider multi-period problems and introduce time indices $t \in \mathcal{T} := \{0, \dots, T\}$ to relevant quantities.

A OPF element e is given as a set of variables $\mathcal{V}_e$, constraints $\mathcal{F}_e$, complex power injection terms S_e^t and an objective function term F_e. Together, all OPF elements form a set $\mathcal{E}$ of size E. The resulting optimization problem takes the form

$$\begin{aligned}
\text{minimize } & \sum_{e\in\mathcal{E}} f_e(\{S^t\}_{t\in\mathcal{T}}, \{U^t\}_{t\in\mathcal{T}}, \mathcal{V}_e) \\
\text{subject to } & (S^t, U^t) \in \mathbb{P} && \forall\, t \in \mathcal{T} \\
& S_j^t = \sum_{e\in\mathcal{E}} S_{e;j}^t && \forall\, t \in \mathcal{T} \\
& \bigcup_{e\in\mathcal{E}} \mathcal{F}_e
\end{aligned}$$

where $\mathbb{P} \in \{\mathbb{V}, \widetilde{\mathbb{V}}, \mathbb{M}, \widetilde{\mathbb{M}}, \dots\}$ is any of the previously described formulations of the relaxed power flow equations.

This encapsulation offers flexibility by allowing to choose the network model and individual optional elements independently.

A. Selected OPF Elements

We describe four OPF elements that we use for the benchmark scenarios and the case study. This list is not exhaustive.

1) Fixed Loads (FL): Fixed loads are given by power injection terms given as $S_{\text{FL},j}^t := -s_{L;j}^t \; \forall j \in \mathcal{N}_{\text{FL}}, t \in \mathcal{T}$ where $s_{L;j}^t$ is the complex-valued load at node j at time t.

No cost function term ($f_{\text{FL}} := 0$) is defined for fixed loads. Neither are any additional variables or constraints ($\mathcal{V}_{\text{FL}} := \{\}$, $\mathcal{F}_{\text{FL}} := \{\}$).

2) Voltage Band (VB): To enforce a permissible voltage interval $[\underline{v}, \overline{v}]$ at a subset $\mathcal{N}_{\text{VB}} \subseteq \mathcal{N}$ of nodes we define a set of constraints

$$\mathcal{F}_{\text{VB}} := \{\underline{v}^2 \le U_j^t \le \overline{v}^2 \quad \forall j \in \mathcal{N}_{\text{VB}}, t \in \mathcal{T}\}.$$

together with $f_{\text{VB}} := 0$, $\mathcal{V}_{\text{VB}} := \{\}$ and $S_{\text{VB},j}^t := 0 \; \forall j \in \mathcal{N}_{\text{VB}}, t \in \mathcal{T}$.

3) 'Box'-Generation with Linear Cost (BG): Given a subset of nodes $\mathcal{N}_{\text{BG}} \subseteq \mathcal{N}$ with generation units we can define

$$\begin{aligned}
\mathcal{V}_{\text{BG}} :=& \{P_{\text{BG};j}^t, Q_{\text{BG};j}^t \in \mathbb{R} && \forall j \in \mathcal{N}_{\text{BG}}, t \in \mathcal{T}\} \\
\mathcal{F}_{\text{BG}} :=& \{\underline{p}_j \le P_{j;\text{BG}}^t \le \overline{p}_j, \\
& \quad \underline{q}_j \le Q_{j;\text{BG}}^t \le \overline{q}_j && \forall j \in \mathcal{N}_{\text{BG}}, t \in \mathcal{T}\} \\
S_{\text{BG};j}^t :=& \; P_{\text{BG};j}^t + iQ_{\text{BG};j}^t && \forall j \in \mathcal{N}_{\text{BG}}, t \in \mathcal{T} \\
f_{\text{BG}} :=& \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{N}_{\text{BG}}} \alpha_j P_{\text{BG};j}^t + \beta_j
\end{aligned}$$

where $\underline{p}_j, \overline{p}_j, \underline{q}_j, \underline{q}_j$ are generation limits for active and reactive power and $\alpha_j, \beta_j \in \mathbb{R}$ are cost coefficients of the generator at node j.

4) Optimal PV-Inverter Dispatch (OID): Inverters of rooftop PV installations connected to distribution grids can be modeled in a OPF problem [4] in order to analyze the need for reactive power control and active power curtailment. For this we model PV inverters in terms of their rated power output, minimum power factor constraints and a maximum power output depending on the current solar irradiation. The OPF element is defined as

$$\begin{aligned}
\mathcal{V}_{\text{OID}} :=& \{P_{\text{OID};j}^t, Q_{\text{OID};j}^t \in \mathbb{R} && \forall j \in \mathcal{N}_{\text{OID}}, t \in \mathcal{T}\} \\
\mathcal{F}_{\text{OID}} :=& \{|Q_{\text{OID};j}^t| \le \gamma_j P_{\text{OID};j}^t && \forall j \in \mathcal{N}_{\text{OID}}, t \in \mathcal{T}, \\
& \quad \left\| \begin{matrix} P_{\text{OID};j}^t \\ q_{\text{OID};j}^t \end{matrix} \right\|_2 \le \overline{s}_j && \forall j \in \mathcal{N}_{\text{OID}}, t \in \mathcal{T}, \\
& \quad 0 \le P_{\text{OID};j}^t \le \overline{p}_j^t && \forall j \in \mathcal{N}_{\text{OID}}, t \in \mathcal{T}\} \\
S_{\text{OID};j}^t :=& \; P_{\text{OID};j}^t + iQ_{\text{OID};j}^t && \forall j \in \mathcal{N}_{\text{OID}}, t \in \mathcal{T}
\end{aligned}$$

where γ_j defines the minimum power factor at node j and $\overline{s}_j$ is the rated power of the inverter. Further, $\overline{p}_j^t$ is maximum available active power depending on the solar irradiation at time t. Since PV installations do not incur fuel costs we set $f_{\text{OID}} := 0$.

5) Optimal Storage Placement (OSP): We consider a simple storage placement scheme that optimally allocates a fixed amount of storage capacity $\overline{C}_{\text{tot}}$ to a subset $\mathcal{N}_{\text{OSP}}$ of nodes and optimizes its dispatch [3]. Storage units can only absorb/inject active power. For this we define a separate OPF element as

$$\begin{aligned}
\mathcal{V}_{\text{OSP}} :=& \{P_{\text{OSP};j}^t, C_{\text{OSP};j}^t \in \mathbb{R} && \forall j \in \mathcal{N}_{\text{OSP}}, t \in \mathcal{T}, \\
& \quad \overline{C}_{\text{OSP};j} \in \mathbb{R} && \forall j \in \mathcal{N}_{\text{OSP}}\} \\
\mathcal{F}_{\text{OSP}} :=& \{C_{\text{OSP};j}^t \le \overline{C}_{\text{OSP};j} && \forall j \in \mathcal{N}_{\text{OSP}}, t \in \mathcal{T}, \\
& \quad |P_{\text{OSP};j}^t| \le \omega \overline{C}_{\text{OSP};j} && \forall j \in \mathcal{N}_{\text{OSP}}, t \in \mathcal{T}, \\
& \quad P_{\text{OSP};j}^t = C_{\text{OSP};j}^{(t+1 \bmod T)} - C_{\text{OSP};j}^t \\
& \qquad && \forall j \in \mathcal{N}_{\text{OSP}}, t \in \mathcal{T}, \\
& \quad \sum_{j\in\mathcal{N}_{\text{OSP}}} \overline{C}_{\text{OSP};j} = \overline{C}_{\text{tot}}\} \\
S_{\text{OSP};j}^t :=& \; P_{\text{OSP};j}^t && \forall j \in \mathcal{N}_{\text{OSP}}, t \in \mathcal{T}
\end{aligned}$$

and $f_{\text{OSP}} := 0$. Here $P_{\text{OSP};j}^t$ denotes the active power injection of the storage unit, $C_{\text{OSP};j}^t$ is the storage state and $\overline{C}_{\text{OSP};j}^t$ its capacity. The coefficient ω_{OSP} limits the injection/absorption to a value proportional to the allocated storage capacity at each node.

V. Benchmarks

Our implementation of the above concepts is written in Python 2.7[3] with MOSEK 7.1[4] as the optimization backend. The benchmark scenarios in this section are based on the benchmark cases included in the MATPOWER [11] software package and two real-world Swiss distribution grids (one radial grid with 109 buses and one meshed grid with 99 buses). The experiments were realized on a laptop with 8 GB RAM and a Intel Core i7 dual-core 2.9 GHz CPU.

[3] www.python.org

[4] www.mosek.com

TABLE I
PROBLEM DIMENSIONS OF TEST CASES

Case	N	C	C_2	#Lnk $\widetilde{\mathbb{M}}$	#Lnk $\mathbb{M}*$
IEEE-118	118	109	9	1,662	682
IEEE-300	300	276	88	4,826	1,956
2383wp	2,383	2,310	650	53,165	22,749
3012sp	3,012	2,920	711	68,266	29,270
3120sp	3,120	3,247	865	76,653	21,834
CH-109rad	109	108	108	1,491	497
CH-99mesh	99	96	70	901	321

TABLE II
COMPUTATION TIMES [S]

Case	$\widetilde{\mathbb{V}}$	$\widetilde{\mathbb{M}}$	$\mathbb{M}^*$	$\mathbb{M}^*_{\text{dyn}}$
IEEE-118	2.62	0.35	0.26	0.24
IEEE-300	49.87	1.96	1.21	1.09
2383wp	-	303.59	73.98	61.29
3012sp	-	447.42	84.71	87.36
3375sp	-	433.95	88.16	88.01
CH-109rad	3.47	0.38	0.33	0.03
CH-99mesh	3.32	0.41	0.27	0.16

Where necessary multiple generation units at the same node have been aggregated and as in [1] and [8] a small resistance value of 10^{-5} has been added to line impedances with otherwise zero resistance. The experiments compare the performance of the different relaxations and formulations in a single-period setting. Fixed load, voltage band and box-generation elements are the only optional problem components. Line limits are not enforced.

We compare the four real formulations, i.e. the full relaxation (i.e. $\widetilde{\mathbb{V}}$), the directly decomposed formulation ($\widetilde{\mathbb{M}}$), the pseudo-complex formulation ($\mathbb{M}^*$) and the pseudo-complex formulation with dynamic constraint types ($\mathbb{M}^*_{\text{dyn}}$).

The full relaxation without decomposition could only be solved for grids with up to 300 buses.

Table I summarizes the problem dimensions. The column C denotes the number cliques and C_2 specifies the number of cliques of size 2. Furthermore, the number of linking constraints represented in the last two columns "#Lnk $\widetilde{\mathbb{M}}$" and "#Lnk $\mathbb{M}^*_{\text{dyn}}$" illustrate that the pseudo-complex formulation requires significantly less linking constraints than the full real conversion. Figure 1 and Table II show that this affects computation time particularly for large meshed problems.

From Table II we further realize that the dynamic choice of constraint type is most effective for the real-world distribution grids "CH-109rad" and "CH-99mes", the latter being almost radial since 70 of its 96 cliques are of size 2.

VI. CASE STUDY EXAMPLE

In order to illustrate the extensibility of our implementation we consider the optimal allocation of storage in a distribution grid based a real-world meshed network topology with 99 buses.

We investigate how optimal storage placements change when large amounts of distributed PV generation (using OID) are added the grid.

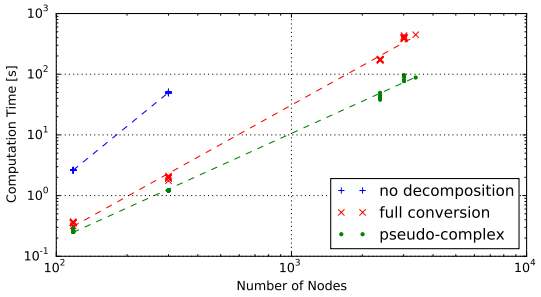


Fig. 1. Computation times for MATPOWER test cases

For this we consider typical load and solar irradiation profiles of a single day with 96 steps (see Figure 2). Realistic simulations would require an optimization over multiple sets of profiles in order to account for meteorological and seasonal variations.

We define a base problem which uses the $\mathbb{M}^*_{\text{dyn}}$ formulation and contains a fixed load element and a voltage band element. Furthermore, a single generation element at the slack node (i.e. the substation) incurs a linear cost on active power generation. Its reactive power generation is unconstrained, active power generation is non-negative. This avoids backfeed of excess PV generation with the goal of maximizing self-sufficiency.

We extend the base problem by adding additional components: A first case applies OSP in the absence of PV generation and a second case considers OSP in the context of high PV penetration. A third scenario considers OID without placing any storage.

Loads are sized according to real consumption data and PV installations are randomly placed in the grid. Their installation size is sampled from a set of typical PV unit dimensions attached to the low-voltage distribution grid.

Table III summarizes the results. In all three cases a voltage vector can be reconstructed from the relaxed optimization problem since the ratio of the largest eigenvalues is close to zero.. As expected, the combined use of PV generation and distributed storage results in the lowest objective value (i.e. the injected active power at the substation) and hence the highest degree of self-sufficiency.

Figure 3 illustrates the allocation of a fixed amount of storage capacity. The storage placement is sparse in the pres-

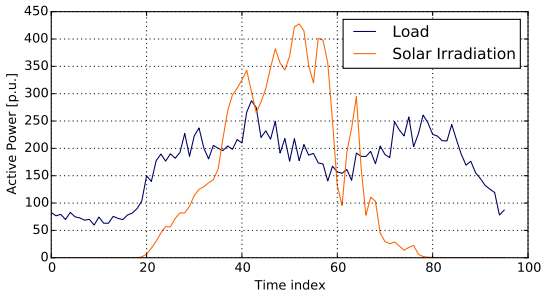
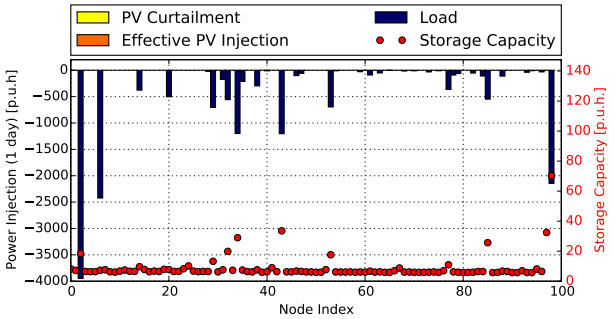
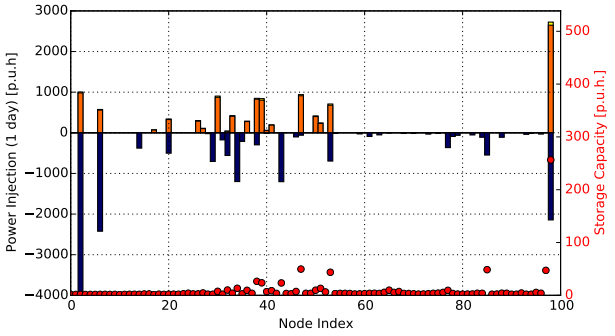


Fig. 2. Aggregated load and solar irradiation profiles

ence of optimally dispatched PV infeed. More elaborate OSP-models could more generally promote sparsity. The allocation itself varies as expected with the presence of PV infeed with PV locations mostly coinciding with storage locations.



(a) Optimal Storage Placement without PV



(b) Optimal Storage Placement and PV-Inverter Dispatch

Fig. 3. OPF scenarios in meshed 109-bus distribution grid

TABLE III
OPTIMAL STORAGE PLACEMENT & INVERTER DISPATCH SCENARIOS

Case	Obj Val	Eig Ratio
OSP, no PV	16,417.7	0.000
PV, no OSP	8,610.8	0.000
OSP & PV	6,228.1	0.001

VII. CONCLUSION

We have developed a computational tool for solving convex relaxations of AC OPF problems. We use chordal conversion methods to exploit problem sparsity. The implementation uses a concise, real-valued problem formulation that avoids introducing redundant constraints. Benchmark experiments show that this feature is particularly effective for the computation of large-scale networks with several thousand buses. Furthermore, the dynamic replacement of 2-by-2 semidefiniteness constraints by second-order cone constraints enables speed-ups by a factor of 10 for radial grids without affecting the tightness of the relaxation or resorting to a different type of problem formulation. Lastly, our software framework uses a modular design that makes it easy to switch between different power flow relaxations and add optional components to the OPF problem in order to compare and evaluate different scenarios.

APPENDIX

A. Proof of Lemma 1

We prove Hermitian symmetry and positive semidefinitness separately. Both $\widetilde{W}^{11}$ and $\widetilde{W}^{22}$ are symmetric by definition. Furthermore, the imaginary part $\widetilde{W}^{12} - \widetilde{W}^{21}$ is antisymmetric since $\widetilde{W}^{12} = (\widetilde{W}^{21})^T$. Hence, W is Hermitian. Positive semidefiniteness of a complex matrix is defined as $z^H W z \geq 0 \; \forall z \in \mathbb{C}^N$. In particular, note that $z^H W z$ has to be real-valued. We write

$$\begin{aligned} z^H W z &= z^H \left(\widetilde{W}^{11} + \widetilde{W}^{22} + i(\widetilde{W}^{12} - \widetilde{W}^{21}) \right) z \\ &= (\Re z)^T \left(\widetilde{W}^{11} + \widetilde{W}^{22} \right) \Re z \\ &\quad + (\Im z)^T \left(\widetilde{W}^{11} + \widetilde{W}^{22} \right) \Im z \\ &\quad + i(\Re z)^T \left(\widetilde{W}^{12} - \widetilde{W}^{21} \right) \Re z \\ &\quad + i(\Im z)^T \left(\widetilde{W}^{12} - \widetilde{W}^{21} \right) \Im z \\ &= (\Re z)^T \left(\widetilde{W}^{11} + \widetilde{W}^{22} \right) \Re z \\ &\quad + (\Im z)^T \left(\widetilde{W}^{11} + \widetilde{W}^{22} \right) \Im z \end{aligned}$$

where the last line follows from the fact that for any antisymmetric real matrix S we have $x^T S x = 0$ for all x. Since both $\widetilde{W}^{11}$ and $\widetilde{W}^{22}$ are psd by Sylvester's criterion we conclude that $W \succeq 0$.

REFERENCES

[1] Martin S. Andersen, Anders Hansson, and Lieven Vandenberghe. Reduced-Complexity Semidefinite Relaxations of Optimal Power Flow Problems. *IEEE Transactions on Power Systems*, 29(4):1855–1863, 2014.

[2] Xiaoqing Bai, Hua Wei, Katsuki Fujisawa, and Yong Wang. Semidefinite programming for optimal power flow problems. *International Journal of Electrical Power & Energy Systems*, 30(6-7):383–392, July 2008.

[3] Subhonmesh Bose, Dennice F. Gayme, Ufuk Topcu, and K. Mani Chandy. Optimal placement of energy storage in the grid. *2012 IEEE 51st IEEE Conference on Decision and Control*, pages 5605–5612, December 2012.

[4] Emiliano Dall'Anese, Sairaj V. Dhople, and Georgios B. Giannakis. Optimal dispatch of photovoltaic inverters in residential distribution systems. *IEEE Transactions on Sustainable Energy*, 5(2):487–497, 2014.

[5] Mituhiro Fukuda, Masakazu Kojima, Murota Kazuo, and Kazuhide Nakata. Exploiting sparsity in semidefinite programming via matrix completion I : General Framework. *SIAM Journal on Optimization*, 11(3):647–674, 2001.

[6] Robert Grone, Charles R Johnson, Eduardo M Sá, and Henry Wolkowicz. Positive definite completions of partial hermitian matrices. *Linear algebra and its applications*, 58:109–124, 1984.

[7] Rabih A. Jabr. Exploiting sparsity in SDP Relaxations of the OPF Problem. *IEEE Transactions on Power Systems*, 27(2):1138–1139, August 2012.

[8] J. Lavaei and S.H. Low. Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems*, 27(1):92–107, Feb 2012.

[9] S.H. Low. Convex Relaxation of Optimal Power Flow - Part I : Formulations and Equivalence. *IEEE Transactions on Control of Network Systems*, (1):15–27, March 2014.

[10] Daniel K. Molzahn, Jesse T. Holzer, Bernard C. Lesieutre, and Christopher L. DeMarco. Implementation of a Large-Scale Optimal Power Flow Solver Based on Semidefinite Programming. *IEEE Transactions on Power Systems*, 28(4):3987–3998, 2013.

[11] R.D. Zimmerman, C.E. Murillo-Sanchez, and R.J. Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *Power Systems, IEEE Transactions on*, 26(1):12–19, Feb 2011.